ppc

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/657,501	09/08/2000	Marc Noel Blais	ROC9-2000-0095-US1	2860

24038 7590 09/11/2003

MARTIN & ASSOCIATES, LLC
P O BOX 548
CARTHAGE, MO 64836-0548

EXAMINER
HOANG, PHUONG N

ART UNIT	PAPER NUMBER
2126	2

DATE MAILED: 09/11/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

P24

Office Action Summary	Application No. 09/657,501	Applicant(s) BLAIS, MARC NOEL	
	Examiner Phuong N. Hoang	Art Unit 2126	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 September 2000* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1 - 41* is/are pending in the application.

4a) Of the above claim(s) *25* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1 - 24, 26 - 41* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *08 September 2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Election/Restrictions

During a telephone conversation with Roy Truelson on 09/04/03 a provisional election was made without traverse to prosecute the invention of Marc Blais. Affirmation of this election must be made by applicant in replying to this Office action. Claim 25 was withdrawn from further consideration by the examiner, 37 CFR 1.142(b) as being drawn to a non-elected invention.

Drawings

The drawings are objected to as failing to comply with 37 CFR 1.84(p)(4) because reference characters "550" and "540" have both been used to designate 540. A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Specification

Applicant refers to the admitted prior art on pages 7 –12, 22, and 23. A copy of the reference is requested by the examiner so that it is fully considered.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

Claims 1 – 4, 6 – 8, 13 – 16, 18 – 20, 26, 29 – 31, 33 - 35, are rejected under 35 U.S.C. 102(a) as being anticipated by Taivalsaari "Implementing a Java Virtual Machine in the Java Programming Language" pages 1 - 23.

As to claim 13, Taivalsaari teaches A method for creating and enforcing protected system level Java code comprising the steps of:

loading a plurality of Java classes (Java classes, p. 5 section 2.2), each of the plurality of Java classes that is protected including state data that indicates a protected class (protected, p. 6 last paragraph);

performing a plurality of checks (These hashtables are used for rapidly finding the various methods and fields of the class at runtime .... protected, private, page 6 last paragraph) when each of the plurality of Java classes is loaded, the plurality of checks determining whether the class being loaded accesses at least one protected class, and if so, determining whether the class being loaded is authorized (execution requirements of each method, page 7 first paragraph) to access the at least one protected class, and generating an exception if the class being loaded is not authorized to access the at least one protected class.

As to claim 14, Taivalsaari teaches the method of claim 13 further comprising the step of performing at least one runtime check when a method that may reference a dynamically defined class (public, protected, private are defined classes in the

hashtables, page 6 last paragraph) is invoked and when a function is invoked that could potentially access a method on one or more of the plurality of classes.

As to claim 15, Taivalsaari teaches at least one runtime check to determine whether a Java reflection (Java reflection, p. 14 paragraph 4, 5 and p. 11 paragraph 1) method is invoked by a referencing class on a referenced class at runtime, and if a Java reflection method is invoked by the referencing class, and if the referenced class implements a private domain interface, and if the referencing class does not implement a system state interface, throwing an exception (exception handlers, page 7 section 2.3);

As to claim 16, Taivalsaari teaches at least one runtime check to determine whether a Java Native Interface (JNI, page 14, section 3.4 – page 15) function is invoked by a program external to the JVM to access a protected class at runtime, and if the program invokes a JNI function to access a protected class, and the program is not running in system state, throwing an exception (exception handlers, page 7 section 2.3);

As to claim 18, 19, Taivalsaari teaches a plurality of checks during class verification that determines whether a class being verified implements a private domain interface (private, page 6 last paragraph) or a system state interface, and if the class being verified is not included in a catalog of allowed classes (the virtual machine creates the corresponding internal runtime structures to represent classes ... access information, p. 5 section 2.2) that is generated during a JVM build process that packages the plurality of classes together into an installable form, throwing an exception (exception handlers, page 7 section 2.3);

As to claim 20, Taivalsaari teaches the plurality of checks during class preparation that determines whether a class being prepared has a superclass (ClassConstantPoolEntry, page 6, paragraph 2), and if the superclass implements a private domain interface (private, page 6 last paragraph) or a system state interface, and if the class being prepared does not implement at least the same private domain interface or system state interface as the superclass, throwing an exception (exception handlers, page 7 section 2.3);

As to claim 26, Taivalsaari teaches:

a state/domain checker that performs a plurality of checks (These hashtables are used for rapidly finding the various methods and fields of the class at runtime .... protected, private, page 6 last paragraph) when each of the plurality of Java classes is loaded, the plurality of checks determining whether the class being loaded accesses at least one protected class, and if so, determining whether the class being loaded is authorized (execution requirements of each method, page 7 first paragraph) to access the at least one protected class, and generating an exception if the class being loaded is not authorized to access the at least one protected class; and

signal bearing media (floppy diskette 195, p. 7) bearing the state/domain checker.

As to claims 29 - 31, see claims 14 - 16 above.

As to claims 33 - 35, see claims 18 – 20 above.

As to claim 1, this is the apparatus claim of claim 13. See claim 13 above.

As to claims 2 – 4, see claims 14 – 16 above.

As to claims 6 - 8, see claims 18 - 20 above.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 5, 17, 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Taivalsaari "Implementing a Java Virtual Machine in the Java Programming Language" pages 1 – 23 in view of Bracha, US patent no. 6,601,114.

As to claim 5, 17, 32, Taivalsaari does not teach the method of claim 13 wherein a class is a protected class if the class is defined as a private domain class or a system state class.

Bracha teaches class is a protected class if the class is defined as a private domain class or a system state class (A private member may be accessed only by members of the class that contains its declaration. A protected member may be accessed by members of the declaring class, col. 3 lines 50 – 60).

It would have been obvious to apply the teaching of Bracha to Taivalsaari's system because the system class needs to be protected from unauthorized accesses.

Claims 9 – 12, 21 – 24, 27, 28, 39 – 41, 36 – 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Taivalsaari "Implementing a Java Virtual

Machine in the Java Programming Language" pages 1 – 23 and in view of the admitted prior art (APA) pages 7 - 12.

As to claim 39, Taivalsaari teaches a program product comprising:

(A) a plurality of Java classes (Java classes, p. 5 section 2.2) at least one of the plurality of Java classes including state data that indicates a protected class (protected, p. 6 last paragraph);

(B) a Java Virtual Machine (JVM) executable program (Java Virtual Machine, p. 5 section 2.2);

(C) a state/domain checker (hashtables, page 6 last paragraph) that performs the following checks:

(C 1) a first check during class verification that determines whether a class being verified implements a private domain interface (private, page 6 last paragraph) or a system state interface, and if the class being verified is not included in a catalog of allowed classes (the virtual machine creates the corresponding internal runtime structures to represent classes ... access information, p. 5 section 2.2) that is generated during a JVM build process that packages the plurality of classes together into an installable form, throwing an exception (exception handlers, page 7 section 2.3);

(C2) a second check during class preparation that determines whether a class being prepared has a superclass (ClassConstantPoolEntry, page 6, paragraph 2), and if the superclass implements a private domain interface (private, page 6 last paragraph) or a system state interface, and if the class being

prepared does not implement at least the same private domain interface or system state interface as the superclass, throwing an exception (exception handlers, page 7 section 2.3);

(C4) a fourth check to determine whether a Java reflection (Java reflection, p. 14 paragraph 4, 5 and p. 11 paragraph 1) method is invoked by a referencing class on a referenced class at runtime, and if a Java reflection method is invoked by the referencing class, and if the referenced class implements a private domain interface, and if the referencing class does not implement a system state interface, throwing an exception (exception handlers, page 7 section 2.3);

(C5) a fifth check to determine whether a Java Native Interface (JNI, page 14, section 3.4 – page 15) function is invoked by a program external to the JVM to access a protected class at runtime, and if the program invokes a JNI function to access a protected class, and the program is not running in system state, throwing an exception (exception handlers, page 7 section 2.3);

Taivalsaari does not explicitly teach:

(C3) a third check during class resolution that determines whether a class being resolved to by a referencing class implements a private domain interface, and if the class being resolved to by the referencing class implements the private domain interface, and if the referencing class does not implement a system state interface, throwing an exception;

(D) signal bearing media bearing the plurality of Java classes, the JVM executable program, and the state/domain checker.

The APA teaches:

(C3) a third check during class resolution (resolution phase, p. 10 lines 12 – p. 11 line 15) that determines whether a class being resolved to by a referencing class implements a private domain interface, and if the class being resolved to by the referencing class implements the private domain interface, and if the referencing class does not implement a system state interface.

(D) signal bearing media (floppy diskette 195, p. 7) bearing the plurality of Java classes, the JVM executable program, and the state/domain checker.

It would have been obvious to apply the teaching of the APA to Taivalsaari's system because the class reference is helpful to transform references in the constant pool into direct references.

As to claim 40, 41, Taivalsaari as modified by the APA teaches the program product of claim 39 wherein said signal bearing media comprises recordable media (APA, floppy diskette 195 is recordable, and transmission media, p. 7).

As to claim 12, this is the apparatus claim of claim 39. See claim 39 above.

As to claim 21, see claim 39(C3) above.

As to claim 22, 23, Taivalsaari modified by the APA teaches the method of claim 21 wherein the check during class resolution is performed at runtime or before runtime when a class is loaded (APA, resolution phase in step 360 may be performed when a

class is loaded, or may instead performed when a method on the class is invoked at runtime, page 11 first paragraph) by a Java Virtual Machine (JVM).

As to claim 24, this is the method claim of claim 39. See claim 39 above.

As to claim 27, 28, see claims 40, 41 above.

As to claim 36 – 38, see claims 21 - 23 above

As to claims 9 – 11, see claims 21 – 23 above.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phuong N. Hoang whose telephone number is (703) 605-4239. The examiner can normally be reached on Monday - Friday 9:00 am to 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Follansbee can be reached on (703)305-8498. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703)746-7140.

Ph

September 5, 2003.

JOHN FOLLANSBEE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100